# CS 402 HW-2

**Name : Anudeep Gadige**
**CWID : A20513979**

**1.5 [4] <§1.6> Consider three diff erent processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.**

| | P1 (GHz) | P2 (GHz) | P3 (GHz) |
|---|---|---|---|
| f (clock rate) | 3 | 2.5 | 4.0 |
| CPI | 1.5 | 1.0 | 2.2 |

a. Which processor has the highest performance expressed in instructions per second?

A) Instructions per second = f / CPI

P1 → $3 \times 10^9 / 1.5 = 2 \times 10^9$
P2 → $2.5 \times 10^9 / 1.0 = 2.5 \times 10^9$
P3 → $4 \times 10^9 / 2.2 = 1.8 \times 10^9$

P2 has the highest performance

b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

A) Given $CPU_{time} = 10s$

Instructions

IC = $(CPU_{time} \times f) / CPI$      IC: Instruction count

P1 → $(10 \times 3 \times 10^9) / 1.5 = 2 \times 10^{10}$ Instructions
P2 → $(10 \times 2.5 \times 10^9) / 1 = 2.5 \times 10^{10}$ Instructions
P3 → $(10 \times 4 \times 10^9) / 2.2 = 1.82 \times 10^{10}$ Instructions

Cycles

Clock cycles = $CPU_{time} \times f$
P1 → $(10 \times 3 \times 10^9) = 3 \times 10^{10}$ cycles
P2 → $(10 \times 2.5 \times 10^9) = 2.5 \times 10^{10}$ cycles

P3 $\rightarrow$ (10 x 4 x $10^9$ ) = 4 x $10^{10}$ cycles

c. We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

A) $f_{new}$ = ($T_{old}$ x $CPI_{new}$ x $f_{old}$) / ($CPI_{old}$ x $T_{new}$)  T: $CPU_{time}$

Let us consider the CPU time (execution time ) to be **y** and the new execution time is **0.7y** (because we are trying to reduce the execution time by 30%)

Increase of 20% in the CPI => $CPI_{new}$ = 1.2 x 1.5

P1 $\rightarrow$ (y x (1.2 x 1.5) x 3 x $10^9$ ) / ( 1.5 x 0.7y ) = 5.14 GHz
P2 $\rightarrow$ (y x (1.2 x 1) x 2.5 x $10^9$ ) / ( 1 x 0.7y ) = 4.29 GHz
P3 $\rightarrow$ (y x (1.2 x 2.2) x 4 x $10^9$ ) / ( 2.2 x 0.7y ) = 6.75 GHz

**1.6 [20] <§1.6> Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (class A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2.**
**Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which implementation is faster?**

A)

$CPU_{time}$ = ∑ (IC x CPI) / f  IC: Instruction count, f: clock rate

a. What is the global CPI for each implementation?

b. Find the clock cycles required in both cases.

**<u>P1</u>**

**$CPU_{time}$** = ( $IC_A$ x $CPI_A$ + $IC_B$ x $CPI_B$ + $IC_C$ x $CPI_C$ + $IC_D$ x $CPI_D$ ) / f

= ($10^5$ x 26) / (2.5 x $10^9$)
= 10.4 x $10^{-4}$ s

**Global CPI** = $(CPU_{time} \times f) / IC$

$= (10.4 \times 10^{-4} \times 2.5 \times 10^{9}) / 10^{6}$
$= 2.6$

**Clock Cycles** = $\sum (IC \times CPI) = 10^{5} \times 26 = 2.6 \times 10^{6}$

<u>**P2**</u>

$\mathbf{CPU_{time}} = ( IC_A \times CPI_A + IC_B \times CPI_B + IC_C \times CPI_C + IC_D \times CPI_D ) / f$

$= (10^{5} \times 20) / (3 \times 10^{9})$
$= 6.67 \times 10^{-4}$ s

**Global CPI** = $(CPU_{time} \times f) / IC$

$= (6.67 \times 10^{-4} \times 3 \times 10^{9}) / 10^{6}$
$= 2$

**Clock Cycles** = $\sum (IC \times CPI) = 10^{5} \times 20 = 2 \times 10^{6}$

**P2 implementation is faster ($CPU_{time}$ of P1 > $CPU_{time}$ of P2 ).**

**1.7 [15] <§1.6> Compilers can have a profound impact on the performance of an application. Assume that for a program, compiler A results in a dynamic instruction count of 1.0E9 and has an execution time of 1.1 s, while compiler B results in a dynamic instruction count of 1.2E9 and an execution time of 1.5 s.**

A) Given,

| | Compiler A | Compiler B |
|---|---|---|
| Execution time ( CPU time) T | 1.1s | 1.5s |
| Instruction count (IC) | $10^{9}$ | $1.2 \times 10^{9}$ |
| Frequency (f) | $10^{9}$ | $10^{9}$ |

a. Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.   f  = 1 / clock cycle = 1 / $10^{-9}$ = $10^{9}$

$CPI_A = (f \times T_A) / IC_A$
$= 10^9 \times 1.1 / 10^9$
$= 1.1$

$CPI_B = (f \times T_B) / IC_B$
$= 10^9 \times 1.5 / 1.2 \times 10^9$
$= 1.25$

b. Assume the compiled programs run on two diff erent processors. If the execution times on the two processors are the same, how much faster is the clock of the processor running compiler A's code versus the clock of the processor running compiler B's code?

If $T_A = T_B$

$f_B / f_A = (IC_B \times CPI_B) / (IC_A \times CPI_A)$
$= (1.25 \times 1.2) / 1.1$
$= 1.36$

The clock speed of B is 36 % faster than A.

c. A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?

Given,
$IC = 6 \times 10^8$
$CPI = 1.1$

$T = (IC \times CPI) / f$
$T_{new}$ (execution time) = $(6 \times 10^8 \times 1.1) / 10^9$ = 0.66s

Speedup = $T_A / T_{new}$ = 1.1/ 0.66 = 1.67
Speedup = $T_B / T_{new}$ = 1.5/ 0.66 = 2.27

The new compiler  is 67% faster than compiler A
The new compiler  is 127% faster than the compiler B.

**1.9 Assume for arithmetic, load/store, and branch instructions, a processor has CPIs of 1, 12, and 5, respectively. Also assume that on a single processor a program requires the execution of 2.56E9 arithmetic instructions, 1.28E9 load/store instructions, and 256 million branch instructions. Assume that each processor has a 2 GHz clock frequency.**

**Assume that, as the program is parallelized to run over multiple cores, the number of arithmetic and load/store instructions per processor is divided by 0.7 x p (where p is the number of processors) but the number of branch instructions per processor remains the same.**

**1.9.1 [5] <§1.7> Find the total execution time for this program on 1, 2, 4, and 8 processors, and show the relative speedup of the 2, 4, and 8 processor result relative to the single processor result.**

A)

Given,

$f = 2 \times 10^9$ Hz

| | Arithmetic (A) | Load / Store (LS) | Branch (B) |
|---|---|---|---|
| CPI | 1 | 12 | 5 |
| IC | 2.56E9 | 1.28E9 | 2.56E8 |
| | | | |

Total execution time ($T_T$) $= T_A + T_{LS} + T_B$

<u>When p = 1</u>

$T_T = ((IC_A \times CPI_A) + (IC_{LS} \times CPI_{LS}) + (IC_B \times CPI_B)) / f$

$T_T = 9.6s$

<u>When p=2</u>

$T_T = (((IC_A \times CPI_A) + (IC_{LS} \times CPI_{LS})) / (0.7 \times 2)) + (IC_B \times CPI_B)) / f$

$T_T$ = 7.04s Speedup = 9.6 / 7.04 = 1.36

When p=4

$T_T$ = $(((IC_A \times CPI_A) + (IC_{LS} \times CPI_{LS})) / (0.7 \times 4)) + (IC_B \times CPI_B)) / f$

$T_T$ = 3.84s Speedup = 9.6 / 3.84 = 2.5

When p=8

$T_T$ = $(((IC_A \times CPI_A) + (IC_{LS} \times CPI_{LS})) / (0.7 \times 8)) + (IC_B \times CPI_B)) / f$

$T_T$ = 2.24s Speedup = 9.6 / 2.24 = 4.29

**1.9.2 [10] <§§1.6, 1.8> If the CPI of the arithmetic instructions was doubled, what would the impact be on the execution time of the program on 1, 2, 4, or 8 processors?**

Given,
$f = 2 \times 10^9$ Hz

| | Arithmetic (A) | Load / Store (LS) | Branch (B) |
|---|---|---|---|
| CPI | 1 | 12 | 5 |
| IC | 2.56E9 x 2 | 1.28E9 | 2.56E8 |
| | | | |

Total execution time ($T_T$) = $T_A + T_{LS} + T_B$

When p = 1

$T_T$ = $((IC_A \times CPI_A) + (IC_{LS} \times CPI_{LS}) + (IC_B \times CPI_B)) / f$

$T_T$ = 10.88s

When p=2

$T_T$ = $(((IC_A \times CPI_A) + (IC_{LS} \times CPI_{LS})) / (0.7 \times 2)) + (IC_B \times CPI_B)) / f$

$T_T$ = 7.954s Speedup = 9.6 / 7.954 = 1.2

When p=4

$T_T$ = $(((IC_A \times CPI_A) + (IC_{LS} \times CPI_{LS})) / (0.7 \times 4)) + (IC_B \times CPI_B)) /\ f$

$T_T$ = 4.29s Speedup = 9.6 / 4.29 = 2.23

When p=8

$T_T$ = $(((IC_A \times CPI_A) + (IC_{LS} \times CPI_{LS})) / (0.7 \times 8)) + (IC_B \times CPI_B)) /\ f$

$T_T$ = 2.46s Speedup = 9.6 / 2.46= 3.9

**1.9.3 [10] <§§1.6, 1.8> To what should the CPI of load/store instructions be reduced in order for a single processor to match the performance of four processors using the original CPI values?**

A) Performance of four processors (when p=4) = 3.84s
$CPI_{new}$ : New CPI of load/ store instructions

$3.84 = ((IC_A \times CPI_A) + (IC_{LS} \times CPI_{new}) + (IC_B \times CPI_B)) /\ f$

$3.84 = ((2.56E9 \times 1) + (1.28E9 \times CPI_{new}) + (5 \times 2.56E8)) / 2E9$

$CPI_{new} = 3$

**1.11 Th e results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.**

**1.11.1 [5] <§§1.6, 1.9> Find the CPI if the clock cycle time is 0.333 ns.**
A)
Given,
IC = 2.389E12
$CPU_{time}$ = 750s
Clock cycle time ($T_{CLK}$ ) = $0.333 \times 10^{-9}$ s

$CPU_{time} = IC \times CPI \times T_{CLK}$

$750 = 2.389E12 \times CPI \times 333 \times 10^{-12}$
CPI = 0.942
**1.11.2 [5] <§1.9> Find the SPECratio.**
A)
SPEC ratio = Reference time / execution time = 9650 /750 = 12.87

**1.11.3 [5] <§§1.6, 1.9> Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% without affecting the CPI.**

A)
Given,
$IC_{new} = 1.1 \times 2.389E12$
$CPU_{time} = IC_{new} \times CPI \times T_{CLK}$
$= 1.1 \times 2.389E12 \times 0.942 \times 0.333 \times 10^{-9}$
$= 824.33s$

Increase in $CPU_{time}$ = 824.33 - 750 = 74.33s

**1.11.4 [5] <§§1.6, 1.9> Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% and the CPI is increased by 5%.**

A)
Given,

$IC_{new} = 1.1 \times 2.389E12$
$CPI_{new} = 1.05 \times 0.942$

$CPU_{time} = IC_{new} \times CPI_{new} \times T_{CLK}$
$= 1.1 \times 2.389E12 \times 1.05 \times 0.942 \times 0.333 \times 10^{-9}$
$= 865.55s$
Increase in $CPU_{time}$ = 865.55 - 750 = 111.55s

**1.11.5 [5] <§§1.6, 1.9> Find the change in the SPECratio for this change.**

A)
SPEC ratio $_{before}$ = Reference time / execution time = 9650 / 824.33 = 11.70
SPEC ratio $_{after}$ = Reference time / execution time = 9650 / 865.55 = 11.14
Change % = (11.70 - 11.14 / 11.70) x 100 = 4.7

**1.11.6 [10] <§1.6> Suppose that we are developing a new version of the AMD Barcelona processor with a 4 GHz clock rate. We have added some additional**

**instructions to the instruction set in such a way that the number of instructions has been reduced by 15%. The execution time is reduced to 700 s and the new SPECratio is 13.7. Find the new CPI.**

A)
Given,
$f = 4E9$ Hz
$IC_{new} = 0.75 \times 2.389E12$
$CPU_{time} = 700s$
SPECratio = 13.7

$CPU_{time} = IC \times CPI_{new} / f$
$700 = (0.85 \times 2.389E12 \times CPI_{new})/ 4E9$
$CPI_{new} = 1.378$

**1.11.7 [10] <§1.6> This CPI value is larger than obtained in 1.11.1 as the clock rate was increased from 3 GHz to 4 GHz. Determine whether the increase in the CPI is similar to that of the clock rate. If they are dissimilar, why?**

A) Frequency ratio = 4/ 3 = 1.33
CPI Ratio = 1.378 / 0.942 = 1.46

CPI increase is dissimilar to that of clock rate, since instructions count and $CPU_{time}$ are reduced.

**1.11.8 [5] <§1.6> By how much has the CPU time been reduced?**
A) Reduction % = $((CPU_{time\ old} - CPU_{time\ new}) / CPU_{time\ old}) \times 100$
= (750-700 / 750 ) x 100
= 6.67 %

**1.11.9 [10] <§1.6> For a second benchmark, libquantum, assume an execution time of 960 ns, CPI of 1.61, and clock rate of 3 GHz. If the execution time is reduced by an additional 10% without affecting to the CPI and with a clock rate of 4 GHz, determine the number of instructions.**

A)
$IC = CPU_{time} \times f / CPI$
$= 960 \times 10^{-9} \times 4E9 / 1.61$
= 2385

$IC_{new} = (CPU_{time\ new} \times f_{new})/(CPI)$

$IC_{new} = (0.9 \times 960 \times 10^{-9} \times 4E9) / (1.61)$

$IC_{new} = 2147$

**1.11.10 [10] <§1.6> Determine the clock rate required to give a further 10% reduction in CPU time while maintaining the number of instructions and with the CPI unchanged.**

A) $f = IC \times CPI / CPU_{time}$
$CPI_{new} = 1.61$
$CPU_{time\ new} = (0.9 \times 960 \times 10^{-9})s$
$IC = 2147$

$f = 2147 \times 1.61 / 777.6 \times 10^{-9}$
$= 4.45E9$ Hz

**1.11.11 [10] <§1.6> Determine the clock rate if the CPI is reduced by 15% and the CPU time by 20% while the number of instructions is unchanged.**

A)
$CPI_{new} = 0.85 \times 1.61$
$CPU_{time\ new} = 0.8 \times 777.6 \times 10^{-9}$ s
$IC = 2147$

$f = 2147 \times 1.368 / 0.8 \times 777.6 \times 10^{-9}$
$= 4.72E9$ Hz

**1.12 Section 1.10 cites as a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider the following two processors. P1 has a clock rate of 4 GHz, average CPI of 0.9, and requires the execution of 5.0E9 instructions. P2 has a clock rate of 3 GHz, an average CPI of 0.75, and requires the execution of 1.0E9 instructions.**

Given,

| | P1 | P2 |
|---|---|---|
| Clock rate (f) | 4GHz | 3GHz |
| CPI | 0.9 | 0.75 |

| | | |
|---|---|---|
| IC | 5E9 | 1E9 |

**1.12.1 [5] <§§1.6, 1.10> One usual fallacy is to consider the computer with the largest clock rate as having the largest performance. Check if this is true for P1 and P2.**

A)

For P1

$CPU_{time}$ = IC x CPI / f
= 5E9 x 0.9 / 4E9
= 1.125s

For P2

$CPU_{time}$ = IC x CPI / f
= 1E9 x 0.75 / 3E9
= 0.25s

Lower CPU times => high performance (CPU time and performance are inversely proportional )

Comparing performance of both P1 and P2, the given statement is false. Largest clock rate doesn't always have the largest performance.

**1.12.2 [10] <§§1.6, 1.10> Another fallacy is to consider that the processor executing the largest number of instructions will need a larger CPU time. Considering that processor P1 is executing a sequence of 1.0E9 instructions and that the CPI of processors P1 and P2 do not change, determine the number of instructions that P2 can execute in the same time that P1 needs to execute 1.0E9 instructions.**

A)
Given,
For P1:

IC = 1E9
f = 4GHz
CPI = 0.9

$CPU_{time}$ = IC x CPI / f
= 1E9 x 0.9 / 4E9
= 0.225 s

For P2:

$CPU_{time}$ = 0.225s
f = 3GHz
CPI = 0.75

IC = $CPU_{time}$ x f / CPI
= 0.225 x 3E9 / 0.75
= 9E8
IC of P2 < IC of P1
Comparing both IC's. The given statement is false, processor executing the largest number of instructions will not always need a larger CPU time

**1.12.3 [10] <§§1.6, 1.10> A common fallacy is to use MIPS (millions of instructions per second) to compare the performance of two different processors, and consider that the processor with the largest MIPS has the largest performance. Check if this is true for P1 and P2.**

A) For P1:
MIPS = f / ($10^6$ x CPI)
= 4E9 / (1E6 x 0.9)
= 4444.45

For P2:
MIPS = f / ($10^6$ x CPI)
= 3E9 / (1E6 x 0.75)
= 4000

Although the MIPS for P1 is high, the performance of P2(lower CPU times) is better than P1. Hence, the given statement is false.

**1.12.4 [10] <§1.10> Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as**
**MFLOPS = No. FP operations / (execution time × 1E6)**
**but this figure has the same problems as MIPS. Assume that 40% of the instructions executed on both P1 and P2 are floating-point instructions. Find the MFLOPS figures for the programs.**

A)
For P1:
Number of FP operations = 0.4 x 5E9

$CPU_{time}$ = IC x CPI / f

= 5E9 x 0.9 / 4E9
= 1.125 s
MFLOPS = 0.4 x 5E9 / (1.125 x 1E6)
= 1777.78

For P2:

Number of FP operations = 0.4 x 1E9

$CPU_{time}$ = IC x CPI / f
= 1E9 x 0.75 / 3E9
= 0.25 s
MFLOPS = 0.4 x 1E9 / (0.25 x 1E6)
= 1600

**1.13 Another pitfall cited in Section 1.10 is expecting to improve the overall performance of a computer by improving only one aspect of the computer. Consider a computer running a program that requires 250 s, with 70 s spent executing FP instructions, 85 s executed L/S instructions, and 40 s spent executing branch Instructions.**

Given,

Time required = 250s
$T_{fp}$ = 70s
$T_{ls}$ = 85s
$T_b$ = 40s
Other operations time = 55s

**1.13.1 [5] <§1.10> By how much is the total time reduced if the time for FP operations is reduced by 20%?**

A)

$T_{fp\ new}$ = 70 x 0.8

Total time = $T_{fp\ new}$ + $T_{ls}$ + $T_b$ + Other operations time= 56 + 85 +40 + 55 = 236s

Reduction % = ((250-236)/ 250) x100 = 5.6

**1.13.2 [5] <§1.10> By how much is the time for INT operations reduced if the total time is reduced by 20%?**

A) Total time (new) = 250 x 0.8 = 200s
$T_{int}$ = Total time - ($T_{fp}$ + $T_{ls}$ + $T_b$ ) = 200 - (70 + 85 +40) = 5s

Integer time reduction % = (55- 5 / 55) x 100 = 90.9

**1.13.3 [5] <§1.10> Can the total time can be reduced by 20% by reducing only the time for branch instructions?**

A) Total time (new) = 250 x 0.8 = 200s
If we avoid branch instructions we could in total time calculation then,
Total time = $T_{fp}$ + $T_{ls}$ + Integer time = 210s
Total time (new) < Total time (without branch time)

Even if we reduce branch instructions time, the total time without the branch instructions is still more than total time reduced by 20%. Hence, the total time cannot be reduced by 20 % by reducing only branch instructions.

**1.14 Assume a program requires the execution of 50 × 106 FP instructions, 110 × 106 INT instructions, 80 × 106 L/S instructions, and 16 × 106 branch instructions. The CPI for each type of instruction is 1, 1, 4, and 2, respectively. Assume that the processor has a 2 GHz clock rate.**

Given,
f = 2GHz

| | FP | INT | L/S | Branch |
|---|---|---|---|---|
| IC | 5E7 | 1.1E8 | 8E7 | 1.6E7 |
| CPI | 1 | 1 | 4 | 2 |
| | | | | |

**1.14.1 [10] <§1.10> By how much must we improve the CPI of FP instructions if Do we want the program to run two times faster?**

A)

$CPU_{time}$ = (IC x CPI) / f
Total $CPU_{time}$ = $CPU_{time\ FP}$ + $CPU_{time\ INT}$ + $CPU_{time\ LS}$ + $CPU_{time\ Branch}$
= ((IC x CPI)$_{fp}$+ (IC x CPI)$_{int}$+ (IC x CPI)$_{ls}$+ (IC x CPI)$_{ls}$)/ f
= 0.256s

Total $CPU_{time\ new}$ = 0.256 / 2 = 0.128s
0.128 = ((IC x $CPI_{new}$)$_{fp}$+ (IC x CPI)$_{int}$+ (IC x CPI)$_{ls}$+ (IC x CPI)$_{ls}$)/ f
$CPI_{new\ fp}$ = (2.56E8 - 4.62E8) / 5E7
= - 2.06E8
Not possible, the CPI cannot be negative.

**1.14.2 [10] <§1.10> By how much must we improve the CPI of L/S instructions if we want the program to run two times faster?**

A) $CPU_{time} = (IC \times CPI) / f$

Total $CPU_{time} = CPU_{time\ FP} + CPU_{time\ INT} + CPU_{time\ LS} + CPU_{time\ Branch}$

$= ((IC \times CPI)_{fp} + (IC \times CPI)_{int} + (IC \times CPI)_{ls} + (IC \times CPI)_{ls}) / f$

$= 0.256s$

Total $CPU_{time\ new} = 0.256 / 2 = 0.128s$

$0.128 = ((IC \times CPI)_{fp} + (IC \times CPI)_{int} + (IC \times CPI_{new})_{ls} + (IC \times CPI)_{ls}) / f$

$CPI_{new\ ls} = (2.56E8 - 1.92E8) / 8E7$

$= 6.4$

**1.14.3 [5] <§1.10> By how much is the execution time of the program improved if the CPI of INT and FP instructions is reduced by 40% and the CPI of L/S and Branch is reduced by 30%?**

A) $CPU_{time} = (IC \times CPI) / f$

Total $CPU_{time} = CPU_{time\ FP} + CPU_{time\ INT} + CPU_{time\ LS} + CPU_{time\ Branch}$

$= ((IC \times CPI)_{fp} + (IC \times CPI)_{int} + (IC \times CPI)_{ls} + (IC \times CPI)_{ls}) / f$

$= ((IC \times CPI \times 0.6)_{fp} + (IC \times CPI \times 0.6)_{int} + (IC \times CPI \times 0.7)_{ls} + (IC \times CPI \times 0.7)_{ls}) / f$

$= 0.189s$

**1.15 [5] <§1.8> When a program is adapted to run on multiple processors in a multiprocessor system, the execution time on each processor is comprised of computing time and the overhead time required for locked critical sections and/or to send data from one processor to another.**
**Assume a program requires t = 100 s of execution time on one processor. When run p processors, each processor requires t/p s, as well as an additional 4 s of overhead, irrespective of the number of processors. Compute the per-processor execution time for 2, 4, 8, 16, 32, 64, and 128 processors. For each case, list the corresponding speedup relative to a single processor and the ratio between actual speedup versus ideal speedup (speedup if there was no overhead).**

A)

When p = 1:

Execution time = 100s

<u>When p = 2</u>:

Execution time = 100 / 2 = 50s

Execution time with overhead = 50 + 4 = 54s
Speedup = 100 / 54 =  1.85
Ideal speedup = 100 / 50 = 2
Speed up/ Ideal speedup = 1.85/2 = 0.925

<u>When p = 4</u>:

Execution time = 100 / 4 = 25s
Execution time with overhead = 25 + 4 = 29s
Speedup = 100 / 29 =  3.44
Ideal speedup = 100 / 25 = 4
Speed up/ Ideal speedup = 3.44/4 = 0.862

<u>When p = 8</u>:

Execution time = 100 / 8 = 12.5s
Execution time with overhead = 12.5 + 4 = 16.5s
Speedup = 100 / 16.5 =  6.06
Ideal speedup = 100 / 12.5 = 8
Speed up/ Ideal speedup = 6.06/8 = 0.757

<u>When p = 16</u>:

Execution time = 100 / 16 = 6.25s
Execution time with overhead = 6.25 + 4 = 10.25s
Speedup = 100 / 10.25 =  9.75
Ideal speedup = 100 / 6.25 = 16
Speed up/ Ideal speedup = 9.75/16 = 0.609